

Sumário/Diário de bordo 07/03/2024

Apresentações do TPC - Exercício de extração dos metadados dos sites: ComUM, NOSUM, UMDicas e entrevistas do projeto Museu da Pessoa.

Objetivos:

- Explicar intenção do código
- Destacar observações importantes
- Registrar observações de alguma forma para o relatório final

Os grupos iniciaram as apresentações dos exercícios começar:

Grupo 2 (Bárbara, Gabriela, Maria Francisca)

ComUm

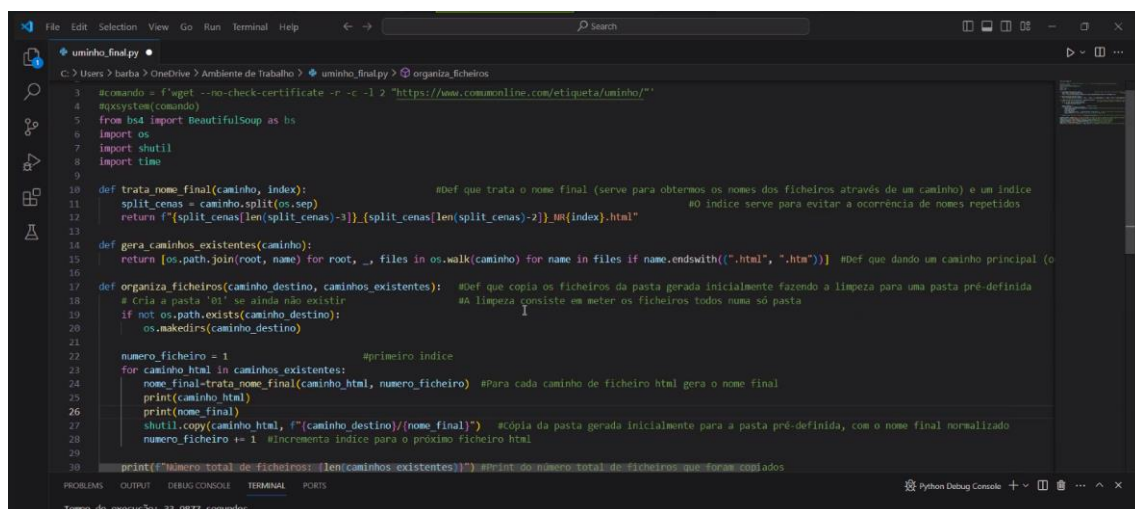
O grupo explicou que realizou a extração das entrevistas por meio da Busca dos artigos por etiquetas: entrevista, Universidade do Minho e Uminho e ao todo extraíram 300 artigos. Fizeram a extração dos artigos que estão em html. Os metadados são diferentes dos utilizados no código para o site NOSUM, por tanto o código inicial não funcionou. As propriedades são parecidas e há algumas citações java. O professor JJ explicou que extensão Aspx é mesma coisa que html e que é possível usar um programa que faz a leitura do aspx - html e converte em html - aspx

O grupo observou que não há diferença no conteúdo entre as etiquetas e que elas auxiliam a encontrar documentos irmãos.

Professor orientou daí pra frente usar:

- `div class="postcontent content">` = buscar o content
- `<h1>` class title = busca o título
- Os códigos do `getartigos.py` pode ser utilizado para buscar os artigos
- aspx: active server page
- Criar pasta com anos, com números e outra com artigos

Códigos utilizados:

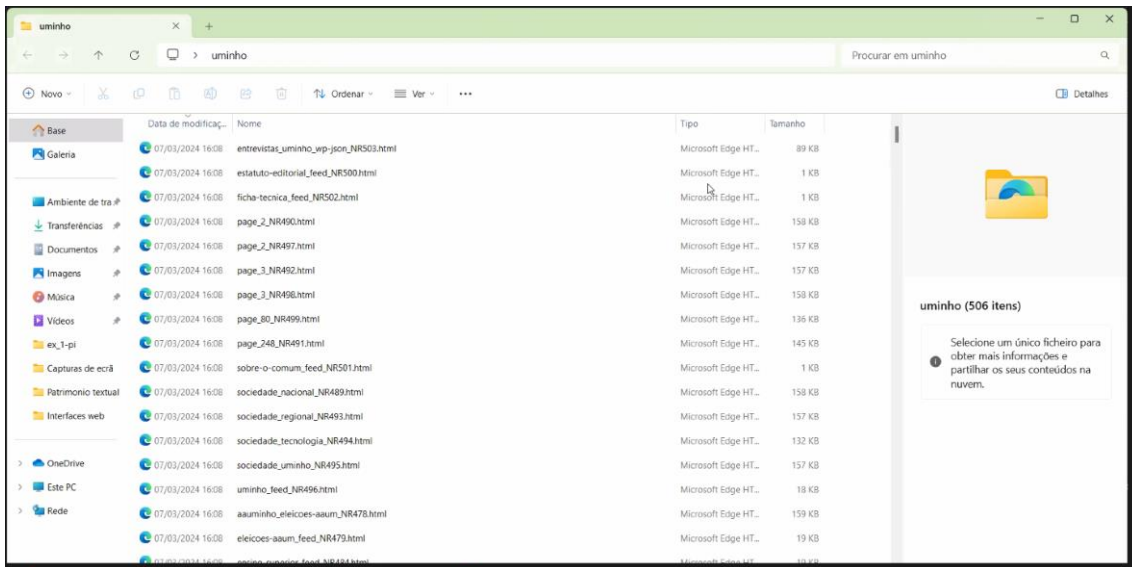


```
1 comando = f'wget --no-check-certificate -r -c -l 2 "https://www.comumonline.com/etiqueta/uminho/"'
2 subprocess.run(comando)
3 from bs4 import BeautifulSoup as bs
4 import os
5 import shutil
6 import time
7
8 def trata_nome_final(caminho, index):
9     #Def que trata o nome final (serve para obtermos os nomes dos ficheiros através de um caminho) e um índice
10     split_cenas = caminho.split(os.sep)
11     # índice serve para evitar a ocorrência de nomes repetidos
12     return f'{split_cenas[len(split_cenas)-3]}_{split_cenas[len(split_cenas)-2]}_{index}.html'
13
14 def gera_caminhos_existentes(caminho):
15     return [os.path.join(root, name) for root, _, files in os.walk(caminho) for name in files if name.endswith((".html", ".htm"))] #Def que dando um caminho principal (o
16
17 def organiza_ficheiros(caminho_destino, caminhos_existentes):
18     #Def que copia os ficheiros da pasta gerada inicialmente fazendo a limpeza para uma pasta pré-definida
19     # Cria a pasta '01' se ainda não existir
20     # A limpeza consiste em meter os ficheiros todos numa só pasta
21     if not os.path.exists(caminho_destino):
22         os.makedirs(caminho_destino)
23
24     numero_ficheiro = 1
25     #primeiro índice
26     for caminho_html in caminhos_existentes:
27         nome_final = trata_nome_final(caminho_html, numero_ficheiro)
28         #Para cada caminho de ficheiro html gera o nome final
29         print(caminho_html)
30         print(nome_final)
31         shutil.copy(caminho_html, f'{caminho_destino}/{nome_final}')
32         #copia da pasta gerada inicialmente para a pasta pré-definida, com o nome final normalizado
33         numero_ficheiro += 1
34         #Incrementa índice para o próximo ficheiro html
35
36 print(f'Número total de ficheiros: {len(caminhos_existentes)}') #Imprimindo número total de ficheiros que foram copiados
37
```

```

28 | numero_ficheiro += 1 #Incrementa indice para o proximo ficheiro html
29 |
30 | print(f"Número total de ficheiros: {len(caminhos_existentes)}") #Print do número total de ficheiros que foram copiados
31 |
32 | caminho_entrada = input("Qual é a pasta de origem? ") #Input que gera a questão, de forma a apenas ser necessário introduzir o nome da que foi gerada inicialmente
33 | tempo_inicio = time.perf_counter() #Obtenção do tempo antes da execução do processo total
34 | caminhos_existentes = gera_caminhos_existentes(caminho_entrada) #Obter a lista de caminhos...
35 | organiza_ficheiros("uminho", caminhos_existentes) #Cópia de ficheiros para a pasta uminho
36 | tempo_fim = time.perf_counter() #Obtenção do tempo depois da execução do processo total
37 | print(f"Tempo de execução: {tempo_fim - tempo_inicio:0.4f} segundos") #Print do tempo (tempo final menos tempo iniciais)
38 |
39 |
40 |

```



Demais passos:

1. Wget (Etiquetas) árvore
2. Arrumação artigos a.html/ Filtrar e selecionar os artigos relacionados a Universidade do Minho
3. Processar cada artigo, retirar o lixo, extrair os metadados, corpo do texto, markdown
4. Criar um script de transformação em markdown

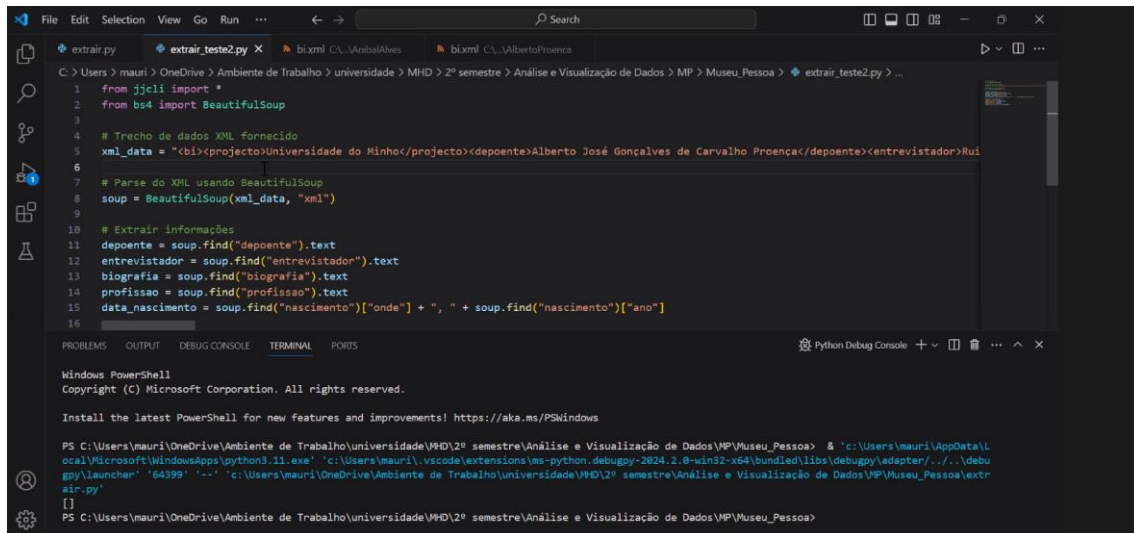
Grupo 4 (Danielle, Mauricio, Renata)

Museu da Pessoa

O grupo realizou a extração das entrevistas salvas no github em xml. E realizou o processamento com wget

Professor informou que o arquivo processado não é xml e está em .json corrompido. Orientou usar o Github desktop e criar um clone do github avd, fazendo um clone especial. Ao fazer o download é feita a cópia na íntegra para não haver distorções.

Códigos usados: pip install lxml (analisador de xml)



```
1 from jjcli import *
2 from bs4 import BeautifulSoup
3
4 # Trecho de dados XML fornecido
5 xml_data = "<bi><projecto>Universidade do Minho</projecto><depoente>Alberto José Gonçalves de Carvalho Proença</depoente><entrevistador>Rui
6
7 # Parse do XML usando BeautifulSoup
8 soup = BeautifulSoup(xml_data, "xml")
9
10 # Extrair informações
11 depoente = soup.find("depoente").text
12 entrevistador = soup.find("entrevistador").text
13 biografia = soup.find("biografia").text
14 profissao = soup.find("profissao").text
15 data_nascimento = soup.find("nascimento")["onde"] + ", " + soup.find("nascimento")["ano"]
16
```

Demais passos:

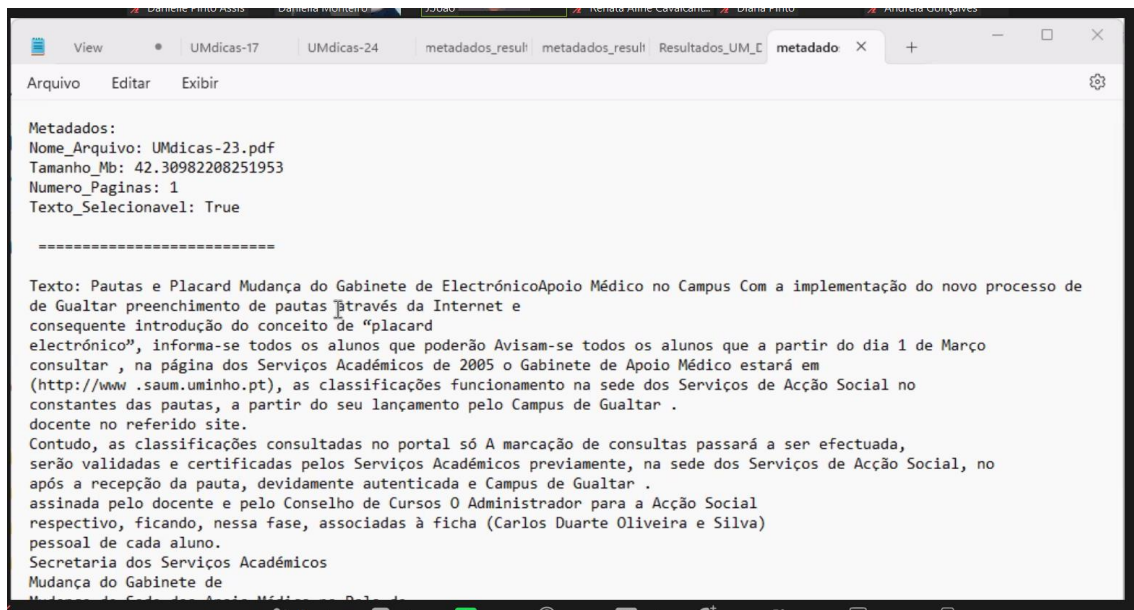
1. Analisar o arquivo entrevista editado
2. Fazer um esquema sobre o que será retirado
3. Arquivos a serem utilizados: Bi.xml, Entrevista editado.xml: sugestão do professor
4. Mapear e distinguir
5. Analisar as etiquetas

Grupo 3 (Cristiana, Daniella, Diana, Lívia)

UMdicas

O Grupo observou que quanto mais antigo o pdf contém mais imagem e menos texto.

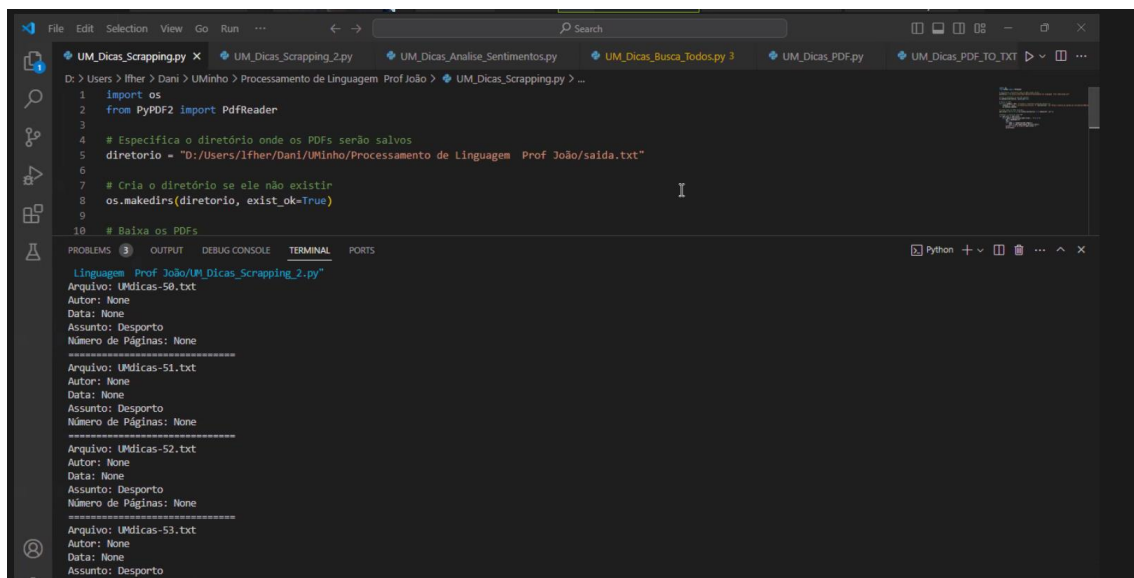
Problemas identificados: como reconhecer o texto que está na imagem e tratar, analisar o texto das entrevistas recentes. Professor sugeriu ignorar as imagens. Grupo conseguiu extrair as datas (analisar formatos das datas), observar se possuem texto ou não, tamanho dos arquivos, Tag reportagem e outras que não têm tag.



Encadeamento realizado:

1. Wget
2. Classificação: data, tamanho, texto, quantidade de páginas
3. Converter pdf para txt 50 entrevistas
4. Verificar se há uma biblioteca que auxilie a delimitação das entrevistas, pois todas terminam como nome do autor
5. Sugestão professor: buscar tag entrevista, conversa com, falar com, pré-marcar para automatizar

Códigos utilizados:



Grupo 1 (Alexandra, Andreia, Francisca)

NOSUM

O grupo explicou que aproveitou os códigos utilizados no arquivo getartigos.py e melhoraram os códigos, os juntando para o processamento de todos os artigos.

Códigos utilizados:

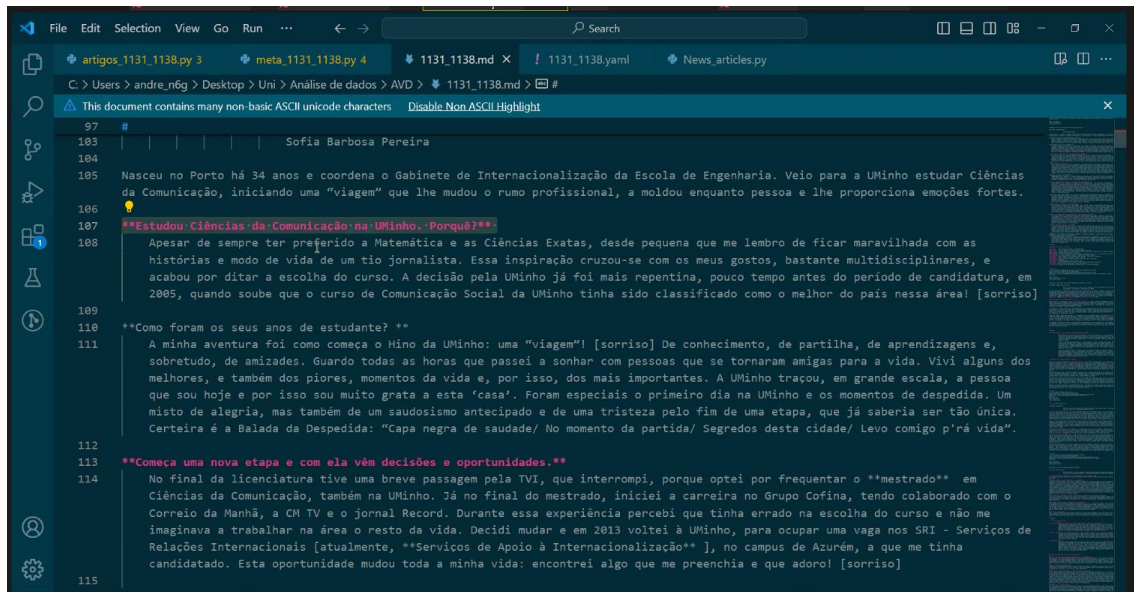
```
File Edit Selection View Go Run ... Search
artigos_1131_1138.py 3 X meta_1131_1138.py 4 1131_1138.md ! 1131_1138.yaml News_articles.py
C:\Users\andre_n6g\Desktop> Uni > Análise de dados > AVD > artigos_1131_1138.py > ...
1 from jicli import *
2 import re
3 from bs4 import BeautifulSoup as bs
4
5
6 ats = glob("1131_1138/Article.aspx*") #Furto do glob para obter uma lista de arquivos que tem nomes começando com "Article.aspx" no
7 print(ats)
8
9 fo = open("1131_1138.md", "w", encoding = "utf-8")
10
11 def proc_article(html):
12     # Processa um ficheiro Article.aspx@id=ldldld do jornal NOSuminho
13     a = bs(html, features="html.parser")
14     cabecalho = ""
15     divisao = "======"
16
17     # Extração dos dados do cabeçalho
18     title = a.find("meta", property="og:title").get("content")
19     url = a.find("meta", property="og:url").get("content")
20     image = a.find("meta", property="og:image").get("content")
21     site_name = a.find("meta", property="og:site_name").get("content")
22     description = a.find("meta", property="og:description").get("content")
23
24     # Extrair data e autor
25     info_span = a.find("span", id="ctl00_ContentPlaceholder1_LabelInfo")
26     date = ""
27     author = ""
28     if info_span:
```

```
File Edit Selection View Go Run ... Search
artigos_1131_1138.py 3 X meta_1131_1138.py 4 1131_1138.md ! 1131_1138.yaml News_articles.py
C:\Users\andre_n6g\Desktop> Uni > Análise de dados > AVD > artigos_1131_1138.py > proc_article
11 def proc_article(html):
12     # Extração dos dados do cabeçalho
13     title = a.find("meta", property="og:title").get("content")
14     url = a.find("meta", property="og:url").get("content")
15     image = a.find("meta", property="og:image").get("content")
16     site_name = a.find("meta", property="og:site_name").get("content")
17     description = a.find("meta", property="og:description").get("content")
18     (variable) info_span: Tag | NavigableString | None
19     info_span = a.find("span", id="ctl00_ContentPlaceholder1_LabelInfo")
20     date = ""
21     author = ""
22     if info_span:
23         match = re.search(r'(\d{2}-\d{2}-\d{4}) \\\ (.+)', info_span.text)
24         if match:
25             date = match.group(1).strip()
26             author = match.group(2).strip()
27
28     # Construir o cabeçalho
29     cabecalho += f"Título: {title}\n"
30     cabecalho += f"URL: {url}\n"
31     cabecalho += f"Imagem: {image}\n"
32     cabecalho += f"Nome do Site: {site_name}\n"
33     cabecalho += f"Descrição: {description}\n"
34     cabecalho += f"Data: {date}\n"
35     cabecalho += f"Autor: {author}\n"
36
37     art = a.find("div", id="artigo")
```

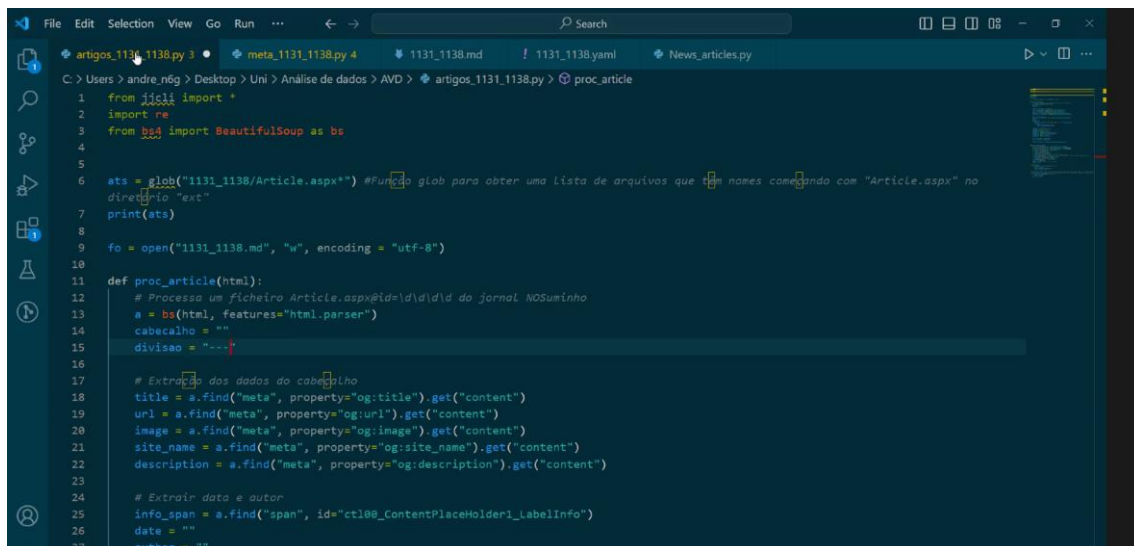
```
File Edit Selection View Go Run ... Search
artigos_1131_1138.py 3 X meta_1131_1138.py 4 1131_1138.md ! 1131_1138.yaml News_artigos.py
C:\Users> andre_n6g > Desktop > Uni > Análise de dados > AVD > artigos_1131_1138.py > proc_article
11 def proc_article(html):
45
46     print("=====\n", cabecalho, divisao, corpo, file = fo)
47
48
49 def proc_art_contents(art):
50     for tag in art.find_all("div", class_="voltar"): tag.extract()
51     for tag in art.find_all("div", id="slidesjs-log"): tag.decompose()
52     for tag in art.find_all("ul", class_="socialcount"): tag.decompose()
53     for tag in art.find_all("div", id="slides"):
54         slides = tag.extract()
55     for tag in art.find_all("table"):
56         tag.extract()
57     for tag in (function) replace_with: Any para cada tag em strong
58         tag.replace_with("***" + tag.text + "***") # substitui por **, deixa o texto, e substitui por **
59     for tag in art.find_all("div", class_="title"):
60         sem_espacos = tag.get_text().strip()
61         tag.replace_with("# " + sem_espacos)
62     finalt = art.get_text()
63     finalt = re.sub(r'\n(3,)', r'\n\n', finalt)
64     return finalt
65
66 for file in ats:
67     with open(file, "r", encoding="utf-8") as f:
68         htmlf = f.read()
69         art = bs(htmlf, 'html.parser')
70         proc_art_contents(art)
71
72 for file in ats: #Itera sobre a lista de arquivos obtida anteriormente. Para cada arquivo, lê o conteúdo HTML, chama a função
```

Resultado:

```
File Edit Selection View Go Run ... Search
artigos_1131_1138.py 3 meta_1131_1138.py 4 1131_1138.md x ! 1131_1138.yaml News_artigos.py
C:\Users> andre_n6g > Desktop > Uni > Análise de dados > AVD > 1131_1138.md > # =====
This document contains many non-basic ASCII unicode characters Disable Non ASCII Highlight
1 =====
2 Título: Natal, pandemia e consumo
3 URL: http://www.nos.uminho.pt/Article.aspx?id=3564
4 Imagem: http://www.nos.uminho.pt/Images/destaques/20211220182345_CludiaSimes.jpg
5 Nome do Site: Jornal Online UMINHO Natal, pandemia e consumo
6 Descrição: Apesar de todo este contexto de incerteza e alguma ansiedade, continuamos, como consumidores, a ser expostos a inúmeros
7 estímulos de marketing apelando ao consumo e a compras de Natal... muitas compras. Afinal, será que algo mudou nos mercados com a
8 pandemia?
9 Data:
10 Autor:
11 =====
12 #
13 Natal, pandemia e consumo
14
15 23-12-2021
16
17 | | | | | Cláudia Simões
18
19 Apesar de todo este contexto de incerteza e alguma ansiedade, continuamos, como consumidores, a ser expostos a inúmeros estímulos de
20 marketing apelando ao consumo e a compras de Natal... muitas compras. Afinal, será que algo mudou nos mercados com a pandemia?
21
22 A época natalícia tem uma magia especial, marcada por símbolos e momentos de celebração e (re)encontros. As casas, as ruas e os
23 espaços comerciais enchem-se de luz, alegria e animação. A árvore de Natal e os presépios entram em muitos lares. Esta quadra é
24 também assinalada pela tradição de dar e receber presentes. Momentos que se traduzem em partilha e satisfação para quem recebe e
25 oferece. No entanto, a situação pandémica que vivemos há quase dois anos traz ao Natal outras tónicas e preocupações. Ocasões com
26 familiares e amigos tornaram-se menos frequentes ou mais restritas e marcadas pela preocupação implícita associada à transmissão da
27 doença Covid 19.
```

```
File Edit Selection View Go Run ... Search
artigos_1131_1138.py 3 meta_1131_1138.py 4 1131_1138.md x ! 1131_1138.yaml News_articles.py
C:\Users\andre_n6g\Desktop> Uni > Análise de dados > AVD > 1131_1138.md > #
This document contains many non-basic ASCII unicode characters Disable Non ASCII Highlight
97 #
103 | | | | Sofia Barbosa Pereira
104
105 Nasceu no Porto há 34 anos e coordena o Gabinete de Internacionalização da Escola de Engenharia. Veio para a UMinho estudar Ciências da Comunicação, iniciando uma "viagem" que lhe mudou o rumo profissional, a moldou enquanto pessoa e lhe proporciona emoções fortes.
106
107 **Estudou Ciências da Comunicação na UMinho. Porque? **
108 Apesar de sempre ter preferido a Matemática e as Ciências Exatas, desde pequena que me lembro de ficar maravilhada com as histórias e modo de vida de um tio jornalista. Essa inspiração cruzou-se com os meus gostos, bastante multidisciplinares, e acabou por ditar a escolha do curso. A decisão pela UMinho já foi mais repentina, pouco tempo antes do período de candidatura, em 2005, quando soube que o curso de Comunicação Social da UMinho tinha sido classificado como o melhor do país nessa área! [sorriso]
109
110 **Como foram os seus anos de estudante? **
111 A minha aventura foi como começa o Hino da UMinho: uma "viagem"! [sorriso] De conhecimento, de partilha, de aprendizagens e, sobretudo, de amizades. Guardo todas as horas que passei a sonhar com pessoas que se tornaram amigas para a vida. Vivi alguns dos melhores, e também dos piores, momentos da vida e, por isso, dos mais importantes. A UMinho traçou, em grande escala, a pessoa que sou hoje e por isso sou muito grata a esta 'casa'. Foram especiais o primeiro dia na UMinho e os momentos de despedida. Um misto de alegria, mas também de um saudosismo antecipado e de uma tristeza pelo fim de uma etapa, que já saberia ser tão única. Certeira é a Balada da Despedida: "Capa negra de saudade/ No momento da partida/ Segredos desta cidade/ Levo comigo p'rá vida".
112
113 **Começa uma nova etapa e com ela vêm decisões e oportunidades.**
114 No final da licenciatura tive uma breve passagem pela TVI, que interrompi, porque optei por frequentar o **mestrado** em Ciências da Comunicação, também na UMinho. Já no final do mestrado, iniciei a carreira no Grupo Cofina, tendo colaborado com o Correio da Manhã, a CM TV e o jornal Record. Durante essa experiência percebi que tinha errado na escolha do curso e não me imaginava a trabalhar na área o resto da vida. Decidi mudar e em 2013 voltei à UMinho, para ocupar uma vaga nos SRI - Serviços de Relações Internacionais [atualmente, **Serviços de Apoio à Internacionalização** ], no campus de Azurém, a que me tinha candidatado. Esta oportunidade mudou toda a minha vida: encontrei algo que me preenchia e que adoro! [sorriso]
115
```

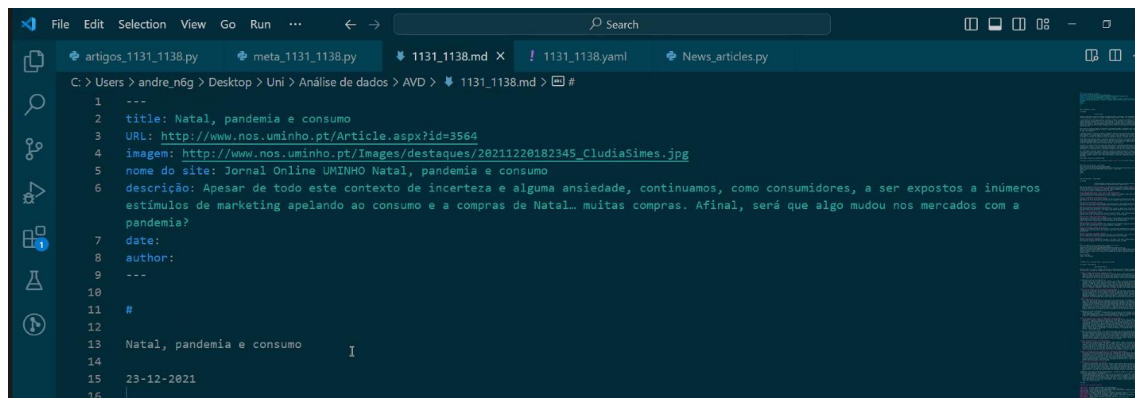


```
File Edit Selection View Go Run ... Search
artigos_1131_1138.py 3 meta_1131_1138.py 4 1131_1138.md ! 1131_1138.yaml News_articles.py
C:\Users\andre_n6g\Desktop> Uni > Análise de dados > AVD > artigos_1131_1138.py > proc_article
1 from urllib import *
2 import re
3 from bs4 import BeautifulSoup as bs
4
5
6 ats = glob("1131_1138/Article.aspx*") #Função glob para obter uma lista de arquivos que têm nomes começando com "Article.aspx" no
diretório "ext"
7 print(ats)
8
9 fo = open("1131_1138.md", "w", encoding = "utf-8")
10
11 def proc_article(html):
12     # Processa um ficheiro Article.aspx?id=1d1d1d do jornal NOSuminho
13     a = bs(html, features="html.parser")
14     cabecalho = ""
15     divisao = "---"
16
17     # Extração dos dados do cabeçalho
18     title = a.find("meta", property="og:title").get("content")
19     url = a.find("meta", property="og:url").get("content")
20     image = a.find("meta", property="og:image").get("content")
21     site_name = a.find("meta", property="og:site_name").get("content")
22     description = a.find("meta", property="og:description").get("content")
23
24     # Extrair data e autor
25     info_span = a.find("span", id="ctl00_ContentPlaceholder1_LabelInfo")
26     date = ""
27     author = ""
```

Professor sugeriu:

- Incluir “”, |---,
- Acertar: title, date, author: para criação do markdown válido
- Na Linha de comando: digitar Pandoc nome ou número do ficheiro.md – s – o _ .html
- Aprender e experimentar criação de arquivo em markdown

Um ficheiro tem vários artigos



```
1 ---
2 title: Natal, pandemia e consumo
3 URL: http://www.nos.uminho.pt/Article.aspx?id=3564
4 imagem: http://www.nos.uminho.pt/Images/destaques/20211220182345_CludiaSimes.jpg
5 nome do site: Jornal Online UMINHO
6 descrição: Apesar de todo este contexto de incerteza e alguma ansiedade, continuamos, como consumidores, a ser expostos a inúmeros
7 estímulos de marketing apelando ao consumo e a compras de Natal... muitas compras. Afinal, será que algo mudou nos mercados com a
8 pandemia?
9 date:
10 author:
11 ---
12 #
13 Natal, pandemia e consumo
14
15 23-12-2021
16
```

TPC:

- Procurar no pandoc markdown sintaxe para ver como aprender
- Experimentar em um texto qualquer incluir tabela, lista, imagem, table of contents para calculo de índice com cardinais

Realizado por: Danielle Assis PG51915